

FILEID**INPUT

M 15

MAC
VO

IIIIII	NN	NN	PPPPPPPP	UU	UU	TTTTTTTTTT
IIIIII	NN	NN	PPPPPPPP	UU	UU	TTTTTTTTTT
II	NN	NN	PP	PP	UU	TT
II	NN	NN	PP	PP	UU	TT
NNNN	NN	NN	PP	PP	UU	TT
NNNN	NN	NN	PP	PP	UU	TT
II	NN	NN	PPPPPPPP	UU	UU	TT
II	NN	NN	PPPPPPPP	UU	UU	TT
II	NN	NNNN	PP	UU	UU	TT
II	NN	NNNN	PP	UU	UU	TT
II	NN	NN	PP	UU	UU	TT
II	NN	NN	PP	UU	UU	TT
II	NN	NN	PP	UUUUUUUUUU	UUUUUUUUUU	TT
II	NN	NN	PP	UUUUUUUUUU	UUUUUUUUUU	TT
IIIIII	NN	NN	PP	UUUUUUUUUU	UUUUUUUUUU	TT
IIIIII	NN	NN	PP	UUUUUUUUUU	UUUUUUUUUU	TT

LL	IIIIII	SSSSSSSS
LL	IIIIII	SSSSSSSS
LL	IIIIII	SS
LLLLLLLL	IIIIII	SSSSSSSS
LLLLLLLL	IIIIII	SSSSSSSS

(2)	73	DECLARATIONS
(3)	102	MAC\$GETCHR GET NEXT CHARACTER FROM INPUT STREAM
(4)	222	MAC\$GETLIN GET NEXT INPUT SOURCE LINE
(5)	324	OPEN NEXT INPUT SOURCE FILE
(6)	359	STAT4 SWITCH INPUT TO MACRO TEXT
(7)	391	MEXIT MACRO EXIT ROUTINE
(7)	403	MAC\$POP INPUT POP INPUT CONTEXT UP A LEVEL
(8)	447	STATEMENTS END-OF-LINE CLEANUP

B C D E F G H I J K L M N O P Q R S T U V W X Y Z

```
0000 1 .TITLE MAC$INPUT GET NEXT CHARACTER
0000 2 .IDENT 'V04-000'
0000 3
0000 4 :
0000 5 :*****
0000 6 :*
0000 7 :* COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0000 8 :* DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0000 9 :* ALL RIGHTS RESERVED.
0000 10 :*
0000 11 :* THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0000 12 :* ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0000 13 :* INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0000 14 :* COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0000 15 :* OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0000 16 :* TRANSFERRED.
0000 17 :*
0000 18 :* THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0000 19 :* AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0000 20 :* CORPORATION.
0000 21 :*
0000 22 :* DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0000 23 :* SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0000 24 :*
0000 25 :*
0000 26 :*****
0000 27 :
0000 28 :
0000 29 :++
0000 30 : FACILITY: VAX MACRO ASSEMBLER OBJECT LIBRARY
0000 31 :
0000 32 : ABSTRACT:
0000 33 :
0000 34 : The VAX-11 MACRO assembler translates MACRO-32 source code into object
0000 35 : modules for input to the VAX-11 LINKER.
0000 36 :
0000 37 : ENVIRONMENT: USER MODE
0000 38 :
0000 39 : AUTHOR: Benn Schreiber, CREATION DATE: 21-AUG-78
0000 40 :
0000 41 : MODIFIED BY:
0000 42 :
0000 43 : V03.01 MTR0011 Mike Rhodes 18-Mar-1982
0000 44 : Add logic to routine MAC$GETLIN to trap non-RMSS EOF errors
0000 45 : and non-SUMS_xxx errors. This occurs when an illegal record
0000 46 : is encountered by RMS and it does not count as a SUM error,
0000 47 : which leaves us in an infinite loop trying to read the next line.
0000 48 : Fixes QAR #691.
0000 49 :
0000 50 : V03.00 MTR0006 Mike Rhodes 15-Mar-1982
0000 51 : Modify MAC$GETCHR to allow ALL characters to be passed
0000 52 : when the FLGSV_DLIMSTR flag is set. The characters allowed
0000 53 : to be passed include the semicolon (which was already passed)
0000 54 : and the hyphen, which at times was incorrectly treated as a
0000 55 : line continuation character. Fixes SPR #11-42904.
0000 56 :
0000 57 : V01.10 RN0023 R. Newland 3-Nov-1979
```

GET NEXT CHARACTER

C 16

16-SEP-1984 02:20:18 VAX/VMS Macro V04-00
5-SEP-1984 01:48:32 [MACRO.SRC]INPUT.MAR;1

Page 2
(1)

0000	58	:	New message codes to get error messages from system message file.
0000	59	:	
0000	60	:	
0000	61	:	V01.09 RN0010 R. Newland 5-Sep-1979
0000	62	:	Mulitpage MXB blocks
0000	63	:	
0000	64	:	V01.08 RN0005 R. Newland 27-Aug-1979
0000	65	:	Remove .ALIGN LONG statements
0000	66	:	
0000	67	:	V01.07 RN0002 R. Newland 01-Feb-1979
0000	68	:	Changes for Source Update Merge, Get input lines
0000	69	:	from SUM\$LINE.
0000	70	:	
0000	71	--	

0000 73 .SBTTL DECLARATIONS
0000 74 : INCLUDE FILES:
0000 75 :
0000 76 :
0000 77 :
0000 78 :
0000 79 : MACROS:
0000 80 :
0000 81 \$RABDEF ;DEFINE RAB OFFSETS
0000 82 \$MAC_GENVALDEF ;VAX-11 MACRO GENERAL SYMBOLS
0000 83 \$MAC_CTLFLGDEF ;CONTROL FLAGS
0000 84 \$MAC_INTCODDEF ;INTERMEDIATE CODE FILE SYMBOLS
0000 85 \$MAC_INPBLKDEF ;DEFINE INPUT BLOCK OFFSETS
003C 86 \$MAC_SYMBLKDEF ;DEFINE SYMBOL BLOCK OFFSETS
0000 87 \$MAC_MNBDEF ;DEFINE MNB OFFSETS
0000 88 \$MACMSGDEF ;Define message codes
0000 89 \$STSDEF ;STATUS BITS
0000 90 DEFSUMCBL ; Define SUM control block symbols
0000 91 :
0000 92 :
0000 93 :
0000 94 : EQUATED SYMBOLS:
0000 95 :
0000 96 :
0000 97 :
0000 98 : OWN STORAGE:
0000 99 :
0000 100 :

0000 102 .SBTTL MAC\$GETCHR GET NEXT CHARACTER FROM INPUT STREAM
 0000 103 :++
 0000 104 : FUNCTIONAL DESCRIPTION:
 0000 105 :
 0000 106 : THIS ROUTINE IS CALLED WHENEVER A NEW CHARACTER IS NEEDED.
 0000 107 : IT PERFORMS ALL THE HOUSEKEEPING FOR ENDS OF LINES, OUTPUTS
 0000 108 : COMMANDS TO THE INT. FILE FOR NEW LINES, AND HANDLES CONTINUATION
 0000 109 : LINES.
 0000 110 :
 0000 111 : CALLING SEQUENCE:
 0000 112 :
 0000 113 JSB MAC\$GETCHR
 0000 114 :
 0000 115 : INPUT PARAMETERS:
 0000 116 :
 0000 117 R10 LAST CHARACTER READ
 0000 118 R11 POINTS TO FLAGS (MAC\$GL FLAGS)
 0000 119 IF FLGSM_ALLCHR IS SET THEN SEMICOLONS WILL BE PASSED
 0000 120 BACK. IF THE FLAG IS CLEAR, SEMICOLONS AND
 0000 121 EVERYTHING FOLLOWING WILL BE IGNORED.
 0000 122 :
 0000 123 : IMPLICIT INPUTS:
 0000 124 :
 0000 125 : NONE
 0000 126 :
 0000 127 : OUTPUT PARAMETERS:
 0000 128 :
 0000 129 R10 NEXT CHARACTER
 0000 130 :
 0000 131 : IMPLICIT OUTPUTS:
 0000 132 :
 0000 133 : NONE
 0000 134 :
 0000 135 : COMPLETION CODES:
 0000 136 :
 0000 137 : NONE
 0000 138 :
 0000 139 : SIDE EFFECTS:
 0000 140 :
 0000 141 : NONE
 0000 142 :
 0000 143 :--
 0000 144 :
 0000 145 :
 0000 146 :
 00000000 147 .PSECT MAC\$RO_CODE_P15,NOWRT,GBL,LONG
 0000 148 :
 0000 149 .ENABL LSB
 0000 150 :
 0000 151 MAC\$GETCHR::
 58 0000'CF 58 DD 0000 152 PUSHL R8 :SAVE R8
 0D 5A 9E 0002 153 10\$: MOVAB W^MAC\$GL_LINEPT,R8 :POINT TO MAC\$GL LINEPT
 1F 12 91 0007 154 20\$: CMPB R10,#CR :TIME TO READ NEW LINE?
 50 5A 20 9A 000C 155 BNEQ 30\$:IF NEQ NO
 6B 08 000F 156 25\$: MOVZBL #BLNK,R10 :YES--PREVENT LOOPING ON CR
 0000'CF 08 C8 0014 157 MOVL W^MAC\$GL_INPUTP_R0 :ADDRESS INPUT BLOCK
 158 BISL2 #FLGSM_CONT,(R11) :ALLOW CONTINUATION LINES

E9	6B	08	B0	16	0017	159	JSB	AINPSL_GETL(R0)	: CALL INPUT ROUTINE TO READ NEXT LINE
		10		E0	001A	160	BBS	#FLGSV_MACTXT,(R11),20\$: BRANCH IF READING MACRO TEXT
					001E	161	\$INTOUT_WD INT\$_NEWL,-		: YES--SIGNAL NEW LINE FOR PASS2
						001E	<W^MAC\$GL_LINENUM>		
5A	00	DC	11	0029	163	30\$:	BRB 20\$: CONTINUE	
		68	D6	002B	164		MOVZBL @R8,R10	: GET NEXT CHARACTER	
		5A	D5	002F	165		INCL (R8)	: POINT TO NEXT CHARACTER	
		F6	13	0031	166		TSTL R10	: WAS CHARACTER A NULL?	
				0033	167		BEQL 30\$: IF EQL YES--GET NEXT CHARACTER	
					0035			: (GOD ONLY KNOWS WHERE NULLS COME FROM!)	
03	6B	2F	E1	0035	169		BBC #FLGSV_DLIMSTR,(R11),35\$: ALLOW SEMICOLONS AND HYPHENS?	
		007B	31	0039	170		BRW 150\$: YES -- BYPASS OTHER TESTS.	
OE	6B	1A	E0	003C	171	35\$:	BBS #FLGSV_SPLALL,(R11),40\$: BRANCH IF SHOULD NOT EVEN CONSIDER	
					0040			: SEMI-COLONS	
3B	5A	91	0040	173			CMPB R10,#SEMI	: IS CHARACTER A SEMI-COLON?	
		09	12	0043	174		BNEQ 40\$: IF NEQ NO	
06	6B	E8	0045	175			BLBS (R11),40\$: YES--AND ARE WE PASSING ALL CHARS.? (ALLCHR IS LOW BIT!!)	
				0048	176				
5A	00	9A	0048	177			MOVZBL #CR,R10	: NO--CALL IT EOL	
	0069	31	004B	178			BRW 150\$: TAKE THE QUICK EXIT	
2D	5A	91	004E	179	40\$:		CMPB R10,#HYPHEN	: LINE CONTINUATION?	
		64	12	0051	180		BNEQ 150\$: IF NEQ NO	
60	6B	03	E1	0053	181		BBC #FLGSV_CONT,(R11),150\$: YES--CONTINUATIONS ALLOWED?	
		5A	DD	0057	182		PUSHL R10	: YES--SAVE CURRENT STATE	
		68	DD	0059	183		PUSHL (R8)	: SAVE MAC\$GL LINEPT	
7E	6B	FFFFFE	8F	CB	005B	184	BICL3 #^C<FLGSM_ALLCHR>,(R11),-(SP)	: SAVE ALLCHR STATE	
		6B	09	CA	0063	185	BICL2 #FLGSM_CONT!FLGSM_ALLCHR,(R11)	: DON'T RECURSE ON LINES THAT HAVE	
		5A	20	9A	0066	186	MOVZBL #BLNK,R10	: ALL HYPHENS.	
		20	5A	91	0069	187	60\$:	CMPB R10,#BLNK	
		05	13	006C	188		BEQL 70\$: IS CHARACTER A BLANK?	
		09	5A	91	006E	189	CMPB R10,#TAB	: IF EQL YES	
		05	12	0071	190		BNEQ 80\$: NO--IS IT A TAB?	
		FF8A	30	0073	191	70\$:	BSBW MAC\$GETCHR	: IF NEQ NO	
		F1	11	0076	192		BRB 60\$: YES--SKIP OVER SPACES AND TABS	
		0D	5A	91	0078	193	80\$:	CMPB R10,#CR	: FIND NON-BLANK,NON-TAB CHARACTER
			2E	12	007B	194		BNEQ 130\$: IS THIS EOL?
		6B	8E	C8	007D	195	BISL2 (SP)+,(R11)	: IF NEQ NO	
		5E	08	CO	0080	196	ADDL2 #2+4,SP	: YES--RESTORE ALLCHR FLAG	
		FF7A	30	0083	197		BSBW MAC\$GETCHR	: AND SAVED CONTEXT NOT NEEDED	
04	6B	0D	E1	0086	198		BBC #FLGSV_OPRND,(R11),90\$: READ NEXT LINE	
06	6B	14	E3	008A	199		BBCS #FLGSV_CHKLPND,(R11),100\$: BRANCH IF NOT IN OPERAND FIELD	
				008E	200	90\$:	SINTOUT_X INT\$_CHKL	: SET CHKL PENDING AND BRANCH IF NONE PEND	
		0D	5A	91	0094	201	100\$:	CMPB R10,#CR	: ENSURE ALIGNMENT OF CONTINUED LINES
		0D	13	0097	202		BEQL 120\$: NULL LINE?	
		0C	5A	91	0099	203	CMPB R10,#FF	: IF EQL YES	
		05	13	009C	204		BEQL 110\$: STILL LOOKING FOR NULL LINES	
		3B	5A	91	009E	205	CMPB R10,#SEMI	:	
		11	12	00A1	206		BEQL 140\$		
		5A	0D	9A	00A3	207	110\$:	MOVZBL #CR,R10	: TREAT AS EOL IF NULL
		FF57	30	00A6	208	120\$:	BSBW MAC\$GETCHR	: READ NEXT LINE	
		E9	11	00A9	209		BRB 100\$: FIND NON-NUL LINE	
				00AB	210	:			
				00AB	211	: NOT REALLY A CONTINUED LINE			
				00AB	212	:			
6B	8E	C8	00AB	213	130\$:	BISL2 (SP)+,(R11)	: RESTORE ALLCHR FLAG		
	68	8ED0	00AE	214		POPL (R8)	: RESTORE MAC\$GL LINEPT		
	5A	8ED0	00B1	215		POPL R10	: RESTORE CHARACTER		

GET NEXT CHARACTER

G 16

MAC\$GETCHR GET NEXT CHARACTER FROM INPUT

16-SEP-1984 02:20:18 VAX/VMS Macro V04-00
5-SEP-1984 01:48:32 [MACRO.SRC]INPUT.MAR;1Page 6
(3)

6B	08	C8	00B4	216	140\$:	BISL2	#FLGSM_CONT,(R11)	:ALLOW CONTINUATIONS AGAIN
58	8ED0	00B7	217	150\$:	POPL	R8	:RESTORE R8	
	05	00BA	218		RSB		:RETURN WITH CHARACTER IN R10	
	00BB	219						
	00BB	220			.DSABL	LSB		

008B 222 .SBTTL MAC\$GETLIN GET NEXT INPUT SOURCE LINE
 008B 223
 008B 224 :++
 008B 225 : THIS ROUTINE IS CALLED TO GET THE NEXT LINE OF THE CURRENT
 008B 226 : INPUT FILE AND PLACE IT IN MAC\$AB_LINEBF.
 008B 227 :--
 008B 228
 008B 229 .ENABL LSB
 008B 230
 008B 231 MAC\$GETLIN::
 50 0000'CF 9E 008B 232 MOVAB W^MAC\$INPUT_RAB, R0 ;POINT TO THE RAB
 0000'CF DF 00C0 233 PUSHAL W^MACSGT_SCB ;Control block address
 00000000'GF 01 FB 00C4 234 CALLS #1,G^SUM\$LINE ;Get next input line
 6D 50 E8 00CB 235 BLBS R0,40\$;If LBS then good read
 00000000'8F 50 D1 00CE 236 CMPL R0,#RMSS_EOF ;Was error end-of-file?
 27 13 00D5 237 BEQL 8\$;Yes if EQL, try next file
 51 50 OC 10 EF 00D7 238 EXTZV #STSSV_FAC_NO,#STS\$S_FAC_NO,R0,R1 ;Get facility no
 0084 8F 51 B1 00DC 239 CMPW R1,#<SUMS_NORMAL@-16\$;SUM error?
 0C 13 00E1 240 BEQL 2\$;Yes if EQL
 51 50 FFFFFFF8 8F CB 00E3 241 BICL3 #^CSTSSM_SEVERITY,R0,R1 ;Copy and check the severity
 CE 13 00EB 242 BEQL MAC\$GETLIN ;WARNING, try again
 OF 11 00ED 243 BRB 8\$;ERROR or FATAL, try next file
 05 6B OE E0 00EF 244 2\$: BBS #FLGSV_P2,(R11),5\$;Return line if Pass 2
 00AF 30 00F3 245 BSBW SUM_ERROR ;Generate intermediate code if Pass 1
 C3 11 00F6 246 BRB MAC\$GETLIN
 0000'CF 3D 07 00F8 247 5\$: DECL W^MACSGL_LINENUM ;Don't increment line number
 3D 11 00FC 248 BRB 40\$
 00FE 249 8\$:
 0000'CF 30 00FE 250 BSBW MAC\$NXTINPFIL ;ELSE TRY TO OPEN NEXT FILE
 0000'CF 01 9A 0101 251 MOVZBL #1,W^MACSGL_SRCPAG ;RESET PAGE COUNT TO ONE
 B2 50 E8 0106 252 BLBS R0,MAC\$GETLIN ;IF THERE IS ANOTHER FILE, GO READ IT
 0000'CF 444E452E 8F D0 0109 253 10\$: MOVL #^A/.END/,W^MAC\$AB_LINEBF ;OOPS--NO FILE--FAKE A '.END'
 0022'CF 04 98 0112 254 MOVZBW #4,W^MAC\$INPUT_RAB+RABSW_RSZ ;WITH A SIZE OF 4 BYTES
 0000'CF D5 0117 255 TSTL W^MACSGL_IF_LEVEL ;IN UNFINISHED CONDITIONAL?
 OC 15 011B 256 BLEQ 30\$;IF LEQ NO
 0004'CF 43 8F 90 011D 258 MOVB #^A/C/,W^MAC\$AB_LINEBF+4;YES--MAKE .END INTO .ENDC
 0022'CF B6 0123 259 INCW W^MAC\$INPUT_RAB+RABSW_RSZ ;COUNT THE CHARACTER
 12 11 0127 260 BRB 40\$;CONTINUE
 OE 6B OE E0 0129 261 30\$: BBS #FLGSV_P2,(R11),40\$;ONLY MESSAGE ON PASS 1
 012D 262 :*** SINTOUT_X INT\$_FN\$NL ;PRINT CONTENTS OF PREVIOUS LINE
 012D 263 SINTOUT_LW INT\$_WRN_- ;TELL OF MISSING END STATEMENT
 012D 264 <#MAC\$_MISSINGEND,#0>
 0022'CF 3C 013B 265 40\$: MOVZWL W^MAC\$INPUT_RAB+RABSW_RSZ,- ;SAVE LENGTH OF LINE
 50 013F 266 R0
 0000'CF 50 D0 0140 267 MOVL R0,W^MACSGL_LINELN ;GET ADDRESS OF LINE BUFFER
 50 0000'CF 9E 0145 268 50\$: MOVAB W^MAC\$AB_LINEBF,R0 ;SET UP LINE POINTER
 0000'CF 50 D0 014A 269 MOVL R0,W^MACSGL_LINEPT ;RESET ERROR POINTER TO LINE START
 0000'CF 50 D0 014F 270 MOVL R0,W^MACSGL_ERRPTX ;NEW SOURCE PAGE?
 OC 60 91 0154 271 CMPB (R0),#FF ;IF NEQ NO
 04 12 0157 272 BNEQ 60\$;YES--COUNT NEW PAGE
 50 0000'CF D6 0159 273 INCL W^MACSGL_SRCPAG ;FIGURE ADDRESS OF LAST CHARACTER
 60 0D 90 0162 274 60\$: ADDL2 W^MACSGL_LINELN,R0 ;STORE CR FOR END OF LINE
 0000'CF D6 0165 275 MOVB #CR,(R0) ;COUNT THIS LINE
 04 6B OE E0 0169 276 INCL W^MACSGL_LINENUM ;BRANCH IF THIS IS PASS 2
 0000'CF D6 016D 277 BBS #FLGSV_P2,(R11),80\$;NO--COUNT LINE READ IN PASS 1
 INCL W^MACSGL_SRC_LCNT

			0171	279	80\$:		
0000'CF	13	6B	51	D4	0171	280	CLRL R1 ; Initialise insert number
	0000'CF	27	E0	0173	281	BBS #FLGSV_UPDFIL,(R11),90\$; Branch if file is being updated	
		50	C3	0177	282	SUBL3 W^MACSGL_LINBAS,- ; Compute line number	
50	17	6B	E1	017F	283	W^MACSGL_LINENUM,R0	
	0000'CF	19	D0	0183	284	BBC #FLGSV_SEQFIL,(R11),100\$; Branch if not sequenced file	
		10	11	0188	285	MOVL W^MACSGL_RECHDBUF,R0 ; Get sequenced line number	
				018A	286	BRB 100\$	
50	0018'CF	3C	018A	287	90\$:	MOVZWL W^MACSGT_SCB+SUM_W_LINE_NO,R0 ; Get original line number	
001C'CF	02	E1	018F	288		BBC #SUM_V_SRCUPD,- ; Branch if line from source file	
		05	0194	289		W^MACSGT_SCB+SUM_B_FLAGS,100\$	
51	001A'CF	3C	0195	290		MOVZWL W^MACSGT_SCB+SUM_W_INSERT_NO,R1 ; Get insert number	
0000'CF	50	80	019A	291	100\$:	MOVW R0,W^MACSGW_LST_LINE ; Save listing line number	
0000'CF	51	80	019F	292		MOVW R1,W^MACSGW_LST_INST ; and insert number	
		05	01A4	293		RSB	
			01A5	294			
			01A5	295			
			01A5	296			
			01A5	297			
			01A5	298			
			01A5	299			
			01A5	300			
0000'CF	04	12	01A5	301			
00	6B	29	E2	01A9	302	TSTL W^MACSGL_LINENUM ; Error on very first line?	
			01AB	303		BNEQ \$S ; No if NEQ	
			01AF	304		BBSS #FLGSV_FIRSTLN,(R11),\$S	
00848800 8F	50	D1	01AF	305	5\$:	CMPL R0,#SUMS_EDITSCLSH ; Edits clash error?	
	22	12	01B6	306		BNEQ 4\$; No if NEQ	
0000001C'EF	03	E0	01B8	307		BBS #SUM_V_SUBCLSH,- ; Branch if not first clash	
	2F		01BF	308		MACSGT_SCB+SUM_B_FLAGS,10\$	
	50	DD	01C0	309		PUSHL R0	
			01C2	310		SINTOUT_X_INTS_FNEWL ; Force new line	
	50	8ED0	01C8	311		POPL R0	
	03	C8	01CB	312		BISL2 #STSSK_INFO,R0 ; Make and information code	
			01CE	313		SINTOUT_LW_INTS_INFO,<R0,#0> ; Information message	
	15	11	01D8	314		BRB 10\$	
			01DA	315	4\$:		
	50	DD	01DA	316		PUSHL R0	
			01DC	317		SINTOUT_X_INTS_FNEWL ; Force new line	
50	8ED0	01E2	01E5	318		POPL R0	
			01EF	319		SINTOUT_LW_INTS_WRN,<R0,#0> ; Warning message	
			01EF	320	10\$:		
			01EF	321		SINTOUT_X_INTS_SUME ; Source update merge error	
	05	01F5	322			RSB	

```

01F6 324 .SBTTL OPEN NEXT INPUT SOURCE FILE
01F6 325
01F6 326 :++
01F6 327 : THIS ROUTINE IS CALLED TO OPEN THE NEXT INPUT FILE.
01F6 328 : IF THE FILE IS SUCCESSFULLY OPENED AND CONNECTED,
01F6 329 : R0 IS SET TO 1. IF ANOTHER INPUT FILE CANNOT BE OPENED,
01F6 330 : R0 IS 0.
01F6 331 :
01F6 332 :--
01F6 333
01F6 334 MAC$NXTINPFIL:
00000000'GF 01 3E 8B 01F6 335 PUSHR #^M<R1,R2,R3,R4,R5> ;SAVE REGISTERS USED BY MOVCS
00000000'GF 01 FB 01F8 336 PUSHAB W^MACSGT SCB ;Supply SUM control block address
00000000'GF 01 0203 337 CALLS #1,G^SUM$CLOSE ;Close any update files
50 0000'CF DO 020E 338 $DISCONNECT RAB=W^MAC$INPUT RAB ;DISCONNECT THE RECORD ACCESS
      0000'CF DO 0213 339 MOVL W^MAC$GL CURINFDB,R0 ;POINT TO CURRENT FDB
      0000'CF DO 021D 340 $CLOSE FAB=8(R0) ;CLOSE THE INPUT FILE
      0000'CF 0221 341 MOVL W^MAC$GL LINENUM,- ;SET LINE BASE TO HIGHEST LINE NUMBER
27 20 6E 00 2C 0224 342 @^MAC$GL LINBAS ;...
      0000'CF 0229 343 MOVC5 #0,(SP),#^A/ /,#39,- ;BLANK FILL THE RESULTANT FILENAME
      0000'CF DO 022C 344 MOVL W^MAC$GL SBT_FILE ;...
0004'CF 50 D1 0231 345 MOVL W^MAC$GL CURINFDB,R0 ;GET ADDRESS OF FDB WE JUST CLOSED
      09 13 0236 346 CMPL R0,W^MAC$GL_INPQUE+4 ;WAS IT THE LAST FDB
      50 60 DO 0238 347 BEQL 20$ ;IF EQL YES
      FDC2' 30 0238 348 MOVL (R0),R0 ;NO--LINK TO NEXT FDB
      3E BA 023E 349 BSBW MAC$OPEN_INPUT ;OPEN NEXT INPUT FILE
      05 0240 350 POPR #^M<R1,R2,R3,R4,R5> ;RESTORE REGISTERS
      0241 351 RSB ;RETURN WITH LBS IN R0
      0241 352 :
      0241 353 : NO MORE INPUT FILES
      0241 354 :
50 D4 0241 355 20$: CLRL R0 ;INDICATE NO MORE FILES
3E BA 0243 356 POPR #^M<R1,R2,R3,R4,R5>
      05 0245 357 RSB

```

0246 359 .SBTTL STAT4 SWITCH INPUT TO MACRO TEXT

0246 360

0246 361 :++

0246 362 : FUNCTIONAL DESCRIPTION:

0246 363 :

0246 364 : THIS SEMANTIC ROUTINE SWITCHES THE POINTER TO THE CURRENT

0246 365 : INPUT BLOCK TO POINT TO THE MACRO BEING EXPANDED. FIRST THE

0246 366 : REAL MACRO ARGUMENTS ARE SCANNED AND AN INPUT BLOCK IS CREATED.

0246 367 : THEN MAC\$GETCHR WILL READ CHARACTERS FROM THE MACRO TEXT

0246 368 : EXPANDER MAC\$GET_MAC_LIN.

0246 369 :

0246 370 :--

0246 371 :

00000000 372 .PSECT MAC\$RO_CODE_P1,NOWRT,GBL,LONG

0000 373 STAT4:: ;STATEMENT = MACTXT

OE 0005'CF E8 0000 374 BLBS W^LSTSG_MACROCALL+SYMSL VAL,10\$;BRANCH IF LISTING MACRO CALLS

56 0000'CF47 D0 0013 375 SINTOUT LW INTS-SETLONG,<#0,#MAC\$GL LIST IT> ;NO--SEND FLAG TO PASS 2

55 00'8F 9A 0019 376 10\$: MOVL W^MAC\$AE VALSTACK[R7],R6 ;GET PTR TO MACRO MNB

FFE0' 30 001D 377 MOVZBL #CRFSK_REF,R5 ;THIS IS A REFERENCE

05 A6 D5 0020 378 BSBW MAC\$CREF MACRO ;CROSS REFERENCE IF CREFPING

1D 13 0023 379 TSTL MNBSL_TXTP(R6) ;IS THERE ANY TEXT

0000'CF 56 D0 0025 380 BEQL 20\$;IF EQL NO--TAKE THE QUICK OUT

0000'CF 18 A6 D0 002A 381 MOVL R6,W^MAC\$GL_MACPTR ;SAVE POINTER TO MACRO MNB

FFCD' 30 0030 382 MOVL MNBSL_ARGP(R6),W^MAC\$GL_KEYMAC ;SET PTR TO KEYWORD ARGS (IF ANY)

0000'CF 0033 383 BSBW MAC\$GET_ARGS ;SCAN REAL MACRO ARGS

0000'CF D4 0033 384 CLRL W^MAC\$GL_KEYMAC ;AND SET UP INPUT BLOCK

0000'CF D0 0037 385 MOVL W^MAC\$GL_BLKPTR,W^MAC\$GL_INPUTP ;CLEAR POINTER TO KEYWORD ARGS

00 6B 10 E3 003E 386 BBCS #FLGSV MACTXT,(R11),20\$;POINT TO NEW INPUT BLOCK

5A 0D 9A 0042 387 MOVZBL #CR,R10 ;FLAG READING MACRO TEXT

05 0045 388 20\$: RSB ;FORCE READING OF NEW LINE

05 0045 389 30\$: RSB

0046 391 .SBTTL MEXIT MACRO EXIT ROUTINE

0046 392

0046 393 :++

0046 394 : FUNCTIONAL DESCRIPTION:

0046 395 :

0046 396 : THIS ROUTINE POPS THE INPUT CONTEXT ONE LEVEL TO EFFECT AN

0046 397 : EXIT FROM A MACRO OR REPEAT-TYPE MACRO

0046 398 :

0046 399 :--

0046 400

0046 401 MEXIT:: :DIRECTIVE = KMEXIT

0046 402

0046 403 .SBTTL MAC\$POP_INPUT POP INPUT CONTEXT UP A LEVEL

0046 404

0046 405 :++

0046 406 : FUNCTIONAL DESCRIPTION:

0046 407 :

0046 408 : THIS ROUTINE POPS THE INPUT CONTEXT A LEVEL. ALL PAGES

0046 409 : ALLOCATED TO THE CURRENT INPUT BLOCK ARE DEALLOCATED.

0046 410 :

0046 411 :--

0046 412

0046 413 MAC\$POP_INPUT::

56 0000'CF 56 DD 0046 414 PUSHL R6 :SAVE R6

1B 6B 10 E1 0048 415 MOVL W^MAC\$GL_INPUTP,R6 :GET POINTER TO CURRENT INPUT BLOCK

50 14 A6 DD 004D 416 BBC #FLGSV_MACTXT,(R11),10\$:BRANCH IF NOT READING MACRO TEXT

15 18 0051 0051 417 MOVL INPSL_RPTCNT(R6),R0 :YES--GET REPEAT COUNT (OR MACRO FLAG)

50 D6 0057 0055 418 BGEQ 10\$:IF GEQ NO NEED TO GO ANY FARTHER

11 13 0059 0057 419 INCL R0 :SEE IF IT WAS REPEAT THAT WENT TO 0

0046 420 BEQL 10\$:IF EQL YES

56 14 A6 80000000 8F CB 005B 421 :NO--IT WAS MACRO REDEFINING ITSELF

FF99' 30 0064 422 BICL3 #^X80000000,INPSL_RPTCNT(R6),R6 :YES--GET ADDRESS OF MNB

56 0000'CF DO 0067 423 BSBW MAC\$DEL_MAC_DEF :AND DELETE THE MNB AND ASSOCIATES

50 66 DO 006C 424 MOVL W^MAC\$GE_INPUTP,R6 :RE-GET THE INPUT BLOCK ADDRESS

0000'CF 50 DO 006F 425 10\$: MOVL INPSL_LINK(R6),R0 :POP INPUT CONTEXT

00000000'8F 50 D1 0074 426 MOVL R0,W^MAC\$GL_INPUTP :READING FROM SOURCE FILE?

04 12 007B 427 CMPL R0,#MAC\$GL_PRMINBL

00 6B 10 E5 007D 428 BNEQ 20\$:IF NEQ NO

0000'CF 10 A6 DO 0081 429 BBC #FLGSV_MACTXT,(R11),20\$:YES--CLEAR MACTXT FLAG

0000'CF 0C A6 DO 0087 430 20\$: MOVL INPSL_IFVAL(R6),W^MAC\$GL :IF VALUE :POP IF VALUE

50 56 DO 008D 431 MOVL INPSL_IFLVL(R6),W^MAC\$GL :IF LEVEL :POP IF LEVEL

56 18 A6 DO 0090 432 MOVL R6,R0 :COPY INPUT BLOCK ADDRESS

FF69' 30 0094 433 MOVL INPSL_PAGP(R6),R6 :LINK TO ANY OTHER PAGES

06 11 0097 434 BSBW MAC\$DEA_1_PAGE :Deallocate block (always 1 page)

0099 435 BRB 40\$:GO DEALLOCATE INPUT BLOCK AND REST

0099 436 :OF INPUT CONTEXT PAGES

56 66 DO 0099 437 30\$: MOVL MXBSL_LINK(R6),R6 : Link to possible next page

FF61' 30 009C 438 BSBW MAC\$DEAL_BLOCK : Deallocate memory block

50 56 DO 009F 439 40\$: MOVL R6,R0 :POINT R0 FOR NEXT DEALLOCATION

F5 12 00A2 441 BNEQ 30\$:IF NEQ GO DO IT

5A 0D 9A 00A4 442 MOVZBL #CR,R10 :FORCE NEW LINE

56 8ED0 00A7 443 POPL R6 :RESTORE R6

05 00AA 444 RSB

	00AB	447	.SBTTL STATEMENTS	END-OF-LINE CLEANUP
	00AB	448		
	0CAB	449	:++	
	00AB	450	:--	AFTER EACH STATEMENT SOME FLAGS NEED TO BE RESET.
	00AB	451		
	00AB	452		
	00AB	453	:MTXT1::	:MARS_TEXT = STATEMENT
	00AB	454	:MTXT2::	:MARS_TEXT = MARS_TEXT STATMENT
	00AB	455	MTXT3::	:MARS_TEXT = DEOL
	00AB	456	MTXT4::	:MARS_TEXT = MARS_TEXT DEOL
	00AB	457	MTEXT:	
	00AB	458		:BEFORE FOR THIS LINE
	00AB	459	\$INTOUT_X INT\$ CHKL	:ALIGN LISTING AND SOURCE
00000046	8F	C8	00B1	BISL2 #FLGSM_BOL!FLGSM_EVALEXPR!FLGSM_COMPEXPR,- ;RESET BEGINNING OF LINE
6B	0000'CF	D4	00B8	0087 460 (R11) ;AND EVALUATE FLAGS
0000'CF	D4	00BC	0088 461 CLRL W^MAC\$GL_ABSFLAG ;RESET ABSOLUTE FLAG	
0000'CF	D4	00C0	00B8 462 CLRL W^MAC\$GL_PRMSEG ;NO EXPRESSION PSECT YET	
6B	01802000	8F	CA	00C4 463 CLRL W^MAC\$GL_MOPPTR ;CLEAR POINTER TO OPERAND MODE BYTES
0000'CF	02	90	00CB	00C4 464 BICL2 #FLGSM_OPRND!FLGSM_IFSTAT!FLGSM_NOREF,(R11) ;NOT IN OPERAND FIELD
0000'CF	02	05	00D0	00CB 465 00CB 466 ;AND NOT IN AN IF
			00D1	00CB 467 ;AND ALLOW PRMSYM TO REF SYMBOLS
			00D1	00D1 468 MOVB #RDX\$V_DECIMAL,W^MAC\$GB_RDXNDX ;RESET TO DECIMAL RADIX
				RSB
				.END

\$\$.TMP1 = 00000001
 \$\$.TMP2 = 000000A0
 \$COUNT = 0000003B
 ARG\$K_SIZE = 000003E8
 AUD\$K_SIZE = 00000010
 BIT... = 00000005
 BLNK = 00000020
 CHRSM_COMMACR = 00000020
 CHRSM_ILL_CRR = 00000040
 CHRSM_NUM_BER = 00000010
 CHRSM_SPA_MSK = 00000001
 CHRSM_SYM_CH1 = 00000008
 CHRSM_SYM_CHR = 00000004
 CHRSM_SYM_DLM = 00000002
 CHRSV_COMMACR = 00000005
 CHRSV_CVTLWC = 00000061
 CHRSV_ILL_CHR = 00000006
 CHRSV_NOCVT = 0000007F
 CHRSV_NUM_BER = 00000004
 CHRSV_SPA_MSK = 00000000
 CHRSV_SYM_CH1 = 00000003
 CHRSV_SYM_CHR = 00000002
 CHRSV_SYM_DLM = 00000001
 CNT = 00000002
 CR = 0000000D
 CRFSK_REF = *****
 ERR = 00000000
 FF = 0000000C
 FLGSM_ALLCHR = 00000001
 FLGSM_BOL = 00000002
 FLGSM_CHKLPND = 00100000
 FLGSM_COMPEXPR = 00000004
 FLGSM_CONT = 00000008
 FLGSM_CRF = 40000000
 FLGSM_CRSEEN = 00000001
 FLGSM_DATRPT = 00000010
 FLGSM_DBGOUT = 00004000
 FLGSM_DLIMSTR = 00008000
 FLGSM_ENDMCH = 00000020
 FLGSM_EVALEXPR = 00000040
 FLGSM_EXPOPT = 00000080
 FLGSM_EXTERR = 00010000
 FLGSM_EXTWRN = 00020000
 FLGSM_FIRSTLN = 00000200
 FLGSM_IFSTAT = 00800000
 FLGSM_IIF = 00400000
 FLGSM_INSERT = 00000100
 FLGSM_IRPC = 20000000
 FLGSM_LEXOP = 00000002
 FLGSM_LSTXST = 00000200
 FLGSM_MAC2COL = 00000800
 FLGSM_MACL = 0C000800
 FLGSM_MACLYB = C8000000
 FLGSM_MACTXT = 00010000
 FLGSM_MEBLST = 00001000
 FLGSM_MOREARG = 00002000
 FLGSM_MOREINP = 00000008

X 04

FLGSM_NEWPND = 00000400
 FLGSM_NOREF = 01000000
 FLGSM_NTYPPEPC = 00000020
 FLGSM_NULCHR = 00040000
 FLGSM_OBJXST = 00200000
 FLGSM_OPNDCHK = 00000100
 FLGSM_OPRND = 00002000
 FLGSM_OPTVFLIDX = 00001000
 FLGSM_ORDLST = 00020000
 FLGSM_P2 = 00004000
 FLGSM_RPTIRP = 10000000
 FLGSM_SEQFIL = 02000000
 FLGSM_SKAN = 00008000
 FLGSM_SPECOP = 00000004
 FLGSM_SPLALL = 04000000
 FLGSM_STOIMF = 00040000
 FLGSM_SYM2COL = 00000400
 FLGSM_TOCFLG = 00080000
 FLGSM_UPAFLG = 00000010
 FLGSM_UPDFIL = 00000080
 FLGSM_UPMARG = 00000040
 FLGSM_XCRF = 80000000
 FLGSV_ALLCHR = 00000000
 FLGSV_BOL = 00000001
 FLGSV_CHKLPND = 00000014
 FLGSV_COMPEXPR = 00000002
 FLGSV_CONT = 00000003
 FLGSV_CRF = C000001E
 FLGSV_CRSEEN = 00000020
 FLGSV_DATRPT = 00000004
 FLGSV_DBGOUT = 0000002E
 FLGSV_DLIMSTR = 0000002F
 FLGSV_ENDMCH = 00000005
 FLGSV_EVALEXPR = 00000006
 FLGSV_EXPOPT = 00000007
 FLGSV_EXTERR = 00000030
 FLGSV_EXTWRN = 00000031
 FLGSV_FIRSTLN = 00000029
 FLGSV_IFSTAT = 00000017
 FLGSV_IIF = 00000016
 FLGSV_INSERT = 00000008
 FLGSV_IRPC = 0000001D
 FLGSV_LEXOP = 00000021
 FLGSV_LSTXST = 00000009
 FLGSV_MAC2COL = 00000028
 FLGSV_MACL = 0000000B
 FLGSV_MACLTB = 0000001B
 FLGSV_MACTXT = 00000010
 FLGSV_MEBLST = 0000000C
 FLGSV_MOREARG = 00000020
 FLGSV_MOREINP = 00000023
 FLGSV_NEWPND = 0000C00A
 FLGSV_NOREF = 00000018
 FLGSV_NTYPPEPC = 00000025
 FLGSV_NULCHR = 00000032
 FLGSV_OBJXST = 00000015
 FLGSV_OPNDCHK = 00000028

FLGSV_OPRND	= 0000000D	INT\$_PRT	= 0000001B
FLGSV_OPTVFLIDX	= 0000002C	INT\$_PSECT	= 0000001C
FLGSV_ORDLST	= 00000011	INT\$_REDEF	= 0000001D
FLGSV_P2	= 0000000E	INT\$_REF	= 0000001E
FLGSV_RPTIRP	= 0000001C	INT\$_REST	= 0000001F
FLGSV_SEQFIL	= 00000019	INT\$_SAME	= 00000009
FLGSV_SKAN	= 0000000F	INT\$_SAVE	= 00000020
FLGSV_SPECOP	= 00000022	INT\$_SBTTL	= 00000021
FLGSV_SPLALL	= 0000001A	INT\$_SETFLAG	= 00000022
FLGSV_STOIMF	= 00000012	INT\$_SETLONG	= 00000023
FLGSV_SYM2COL	= 0000002A	INT\$_SPIC	= 00000024
FLGSV_TOCFLG	= 00000013	INT\$_SPID	= 00000025
FLGSV_UPAFLG	= 00000024	INT\$_STIB	= 00000026
FLGSV_UPDFIL	= 00000027	INT\$_STIL	= 00000028
FLGSV_UPMARG	= 00000026	INT\$_STIW	= 00000027
FLGSV_XCRF	= 0000001F	INT\$_STKEPT	= 00000029
HASHSZ	= 0000007F	INT\$_STKG	= 0000002A
HYPHEN	= 0000002D	INT\$_STKL	= 0000002B
INPSB_ARGCT	= 0000001C	INT\$_STKPC	= 0000002C
INPSK_BLKSIZ	= 00000021	INT\$_STKS	= 0000002D
INPSK_BUFSIZ	= 000003E8	INT\$_STCB	= 00000034
INPSK_IRPSIZ	= 0000003C	INT\$_STOL	= 0000002E
INPSL_ARGS	= 0000001D	INT\$_STOW	= 00000035
INPSL_GETL	= 00000008	INT\$_STRB	= 0000002F
INPSL_IFLVL	= 0000000C	INT\$_STRL	= 00000031
INPSL_IFVAL	= 00000010	INT\$_STRSB	= 00000032
INPSL_LINK	= 00000000	INT\$_STRSW	= 00000033
INPSL_NXTL	= 00000004	INT\$_STRW	= 00000030
INPSL_PAGP	= 00000018	INT\$_STS8	= 00000036
INPSL_RPTCNT	= 00000014	INT\$_STS8	= 00000037
INTSK_BUFSIZ	= 000013F4	INT\$_SUB	= 0000000A
INTSK_BUFWRN	= 00001390	INT\$_SUME	= 00000039
INTS_ADD	= 00000001	INT\$_WRN	= 00000038
INTS_AND	= 00000002	INT\$_XOR	= 0000000B
INTS_ASH	= 00000003	LST\$G_MACROCALL	***** X 04
INTS ASN	= 0000000C	LST\$K_BUFSIZ	= 00000086
INTS AUGPC	= 0000000D	LST\$K_L_P PAGE	= 0000003C
INTS_BDST	= 0000000E	LST\$K_TITLE_SIZ	= 00000028
INTS_CHKLN	= 0000000F	MAB\$B_ARGNO	00000005
INTS_DIV	= 00000004	MAB\$B_NAME	00000004
INTS_END	= 00000010	MAB\$K_BLKSIZ	0000000C
INTS_EPT	= 00000011	MAB\$L_DVPTR	00000008
INTS_ERR	= 00000012	MAB\$L_LINK	00000000
INTS_ETX	= 00000013	MAB\$W_DVLEN	00000006
INTS_FNEWL	= 00000014	MAC\$AB_LINEBF	***** X 03
INTS_ILG	= 00000000	MAC\$AB_SBT_FILE	***** X 03
INTS_INFO	= 0000003A	MAC\$AL_VALSTACK	***** X 04
INTS_LGLAB	= 00000015	MAC\$CREF_MAC 0	***** X 04
INTS_MACL	= 00000016	MAC\$DEAL_BLOCK	***** X 04
INTS_MUL	= 00000005	MAC\$DEA_T PAGE	***** X 04
INTS_NEG	= 00000006	MAC\$DEL_MAC DEF	***** X 04
INTS_NEWL	= 00000017	MAC\$GB_RDXNDX	***** X 04
INTS_NEWP	= 00000018	MAC\$GETCHR	00000000 RG 03
INTS_NOT	= 00000007	MAC\$GETLIN	000000BB RG 03
INTS_OP	= 00000019	MAC\$GET_ARGS	***** X 04
INTS_OR	= 00000008	MAC\$GL_ABSFLAG	***** X 04
INTS_PRIL	= 0000001A	MAC\$GL_BLKPTR	***** X 04

MAC\$INPUT
Symbol table

GET NEXT CHARACTER

D 1

16-SEP-1984 02:20:18 VAX/VMS Macro V04-00
5-SEP-1984 01:48:32 [MACRO.SRC]INPUT.MAR;1

Page 15
(8)

MACSGL_CURINFDB	*****	X	03	PSCSK_BLKSIZ	= 00000013
MACSGL_ERRPTX	*****	X	03	PSCSK_NO_OPTNS	= 0000000A
MACSGL_IF_LEVEL	*****	X	03	PSCSL_CURLOC	= 0000000F
MACSGL_IF_VALUE	*****	X	04	PSCSL_LINK	= 00000000
MACSGL_INPQUE	*****	X	03	PSCSL_MAXLNGTH	= 00000005
MACSGL_INPUTP	*****	X	03	PSCSM_ABS	= FFFFFFF7
MACSGL_KEYMAC	*****	X	04	PSCSM_ALIGNFLG	= 00004000
MACSGL_LINBAS	*****	X	03	PSCSM_ALLOPTNS	= 000003FF
MACSGL_LINELN	*****	X	03	PSCSM_BYTE	= 00004000
MACSGL_LINENUM	*****	X	03	PSCSM_CON	= FFFFFFFB
MACSGL_LINEPT	*****	X	03	PSCSM_DEFAULT	= 000001C8
MACSGL_LIST_IT	*****	X	04	PSCSM_EXE	= 000000C0
MACSGL_MACPTR	*****	X	04	PSCSM_GBL	= 00000010
MACSLL_MOPPTR	*****	X	04	PSCSM_LCL	= FFFFFFFE
MACSGL_PRMINBL	*****	X	04	PSCSM_LIB	= 00000002
MACSGL_PRMSEG	*****	X	04	PSCSM_LONG	= 00004800
MACSGL_RECHDBUF	*****	X	0.	PSCSM_NOEXE	= FFFFFFBF
MACSGL_SRCPAG	*****	X	0.	PSCSM_NOPIC	= FFFFFFFE
MACSGL_SRC_LCNT	*****	X	03	PSCSM_NORD	= FFFFFF7F
MACSGT_SCB	*****	X	03	PSCSM_NOSHR	= FFFFFFDF
MACSGW_LST_INST	*****	X	03	PSCSM_NOVEC	= FFFFFDFF
MACSGW_LST_LINE	*****	X	03	PSCSM_NOWRT	= FFFFFEFF
MACSINPUT_RAB	*****	X	03	PSCSM_OVR	= 00000004
MACSINTERR_2_LW	*****	X	03	PSCSM_PAGE	= 00006400
MACSINTOUT_2_LW	*****	X	03	PSCSM_PIC	= 00000001
MACSINTOUT_WD	*****	X	03	PSCSM_QUAD	= 00004C00
MACSINTOUT_X	*****	X	03	PSCSM_RD	= 00000080
MACSNXTINPFIL	000001F6	R	03	PSCSM_REL	= 00000008
MACSOPEN_INPUT	*****	X	03	PSCSM_SHR	= 00000020
MACSPOP_INPUT	00000046	RG	04	PSCSM_USR	= FFFFFFFD
MACS MISSINGEND	= 007D8828			PSCSM_VEC	= 0000200
MAC SUBSYS	= 0000007D			PSCSM_WORD	= 00004400
MEXIT	00000046	RG	04	PSCSM_WRT	= 0000180
MNBSB_ARGCT	00000017			PSCSS_ALIGNMENT	= 0000004
MNBSB_NAME	00000004			PSCSV_ALIGNFLG	= 000000E
MNBSK_BLKSIZ	0000001C			PSCSV_ALIGNMENT	= 000000A
MNBSL_ARGP	00000018			PSCSV_EXE	= 0000006
MNBSL_CRSYMF	00000013			PSCSV_GBL	= 0000004
MNBSL_LINK	00000000			PSCSV_LIB	= 0000001
MNBSL_PAGC	0000000F			PSCSV_OVR	= 0000002
MNBSL_PAGP	0000000B			PSCSV_PIC	= 0000000
MNBSL_TXTP	00000005			PSCSV_RD	= 0000007
MNBSW_FLAG	00000009			PSCSV_REL	= 0000003
MTEXT	000000AB	R	04	PSCSV_SHR	= 0000005
MTXT3	000000AB	RG	04	PSCSV_VEC	= 0000009
MTXT4	000000AB	RG	04	PSCSV_WRT	= 0000008
MXBHK_BLKSIZ	00000008			PSCSW_FLAG	= 0000009
MXBHK_LINK	00000000			PSCSW_OPTIONS	= 000000D
MXBHK_PAGES	00000004			RABSW_RSZ	= 00000022
OBJSK_BUFSIZ	= 00000200			RDXSV_BINARY	= 0000000
OPFSM_LASTOPR	= 00002000			RDXSV_DECIMAL	= 0000002
OPFSM_OPTEXP	= 00001000			RDXSV_DOUBLE	= 0000005
OPFSV_LASTOPR	= 0000000D			RDXSV_FLOAT	= 0000004
OPFSV_OPTEXP	= 0000000C			RDXSV_GFLOAT	= 0000006
PSCSB_NAME	00000004			RDXSV_HEX	= 0000003
PSCSB_SEG	0000000C			RDXSV_HFLOAT	= 0000007
PSCSB_UNUSED	00000008			RDXSV_OCTAL	= 0000001

MA
Sy

MACSINPUT Symbol table

GET NEXT CHARACTER

E 1

16-SEP-1984 02:20:18 VAX/VMS Macro V04-00
5-SEP-1984 01:48:32 [MACRO.SRC]INPUT.MAR;1

Page 16
(8)

REGS_PC	=	0000000
RMS\$_EOF	=	*****
SEMI	=	0000000
SIZ...	=	0000000
STAT4	=	0000000
STBSK_PG_MISS	=	0000000
STS\$K_INFO	=	0000000
STSSM_SEVERITY	=	0000000
STSS\$_FAC_NO	=	0000000
STSSV_FAC_NO	=	0000000
SUMSCLOSE	=	0000000
SUMSLINE	=	*****
SUMS_EDITSCLSH	=	008488
SUMS_NORMAL	=	008480
SUM_B_FLAGS	=	0000000
SUM_ERROR	=	000001
SUM_K_BLN	=	0000000
SUM_L_ISDATA	=	0000000
SUM_L_STS	=	0000000
SUM_M_AUDIT	=	0000000
SUM_M_AUDITNEW	=	0000000
SUM_M_DELETE	=	0000000
SUM_M_SRCUPD	=	0000000
SUM_M_SUBCLSH	=	0000000
SUM_Q_AUDD\$	=	0000000
SUM_G_FILESP	=	0000000
SUM_V_AUDIT	=	0000000
SUM_V_AUDITNEW	=	0000000
SUM_V_DELETE	=	0000000
SUM_V_SRCUPD	=	0000000
SUM_V_SUBCLSH	=	0000000
SUM_W_INSERT NO	=	0000000
SUM_W_LINE_N0	=	0000000
SYMSB_NAME	=	0000000
SYMSB_SEG	=	0000000
SYMSB_TOKEN	=	0000000
SYMSK_BLKSIZ	=	0000000
SYMSK_MAXLEN	=	0000000
SYMSK_TWOCOL	=	0000000
SYMSL_LINK	=	0000000
SYMSL_VAL	=	0000000
SYMSM_ABS	=	0000000
SYMSM ASN	=	000001
SYMSM_CRF0	=	000020
SYMSM_DEBUG	=	0000000
SYMSM_DEF	=	0000000
SYMSM_DELMAC	=	000002
SYMSM_EPT	=	000002
SYMSM_EXTRN	=	0000000
SYMSM_GLOBL	=	0000000
SYMSM_LOCAL	=	0000000
SYMSM_ODBG	=	000004
SYMSM_REF	=	0000000
SYMSM_RELPECT	=	000008
SYMSM_SUPR	=	000040
SYMSM_WEAK	=	0000000
SYMSM_XCRF	=	000010

```
+-----+
! Psect synopsis !
+-----+
```

PSECT name

. ABS :
. BLANK :
\$ABSS
MAC\$RO_CODE_P15
MAC\$RO_CODE_P1

Allocation

	Allocation	PSECT No.	Attributes
. ABS :	000000000 (0.)	00 (0.)	NOPIC USR CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE
. BLANK :	000000000 (0.)	01 (1.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC BYTE
\$ABSS	00000003C (60.)	02 (2.)	NOPIC USR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE
MAC\$RO_CODE_P15	00000246 (582.)	03 (3.)	NOPIC USR CON REL GBL NOSHR EXE RD NOWRT NOVEC LONG
MAC\$RO_CODE_P1	0000000D1 (209.)	04 (4.)	NOPIC USR CON REL GBL NOSHR EXE RD NOWRT NOVEC LONG

```
+-----+
! Performance indicators !
+-----+
```

Phase

Phase	Page faults	CPU Time	Elapsed Time
Initialization	29	00:00:00.02	00:00:01.67
Command processing	107	00:00:00.48	00:00:04.12
Pass 1	263	00:00:04.89	00:00:24.79
Symbol table sort	0	00:00:00.65	00:00:02.10
Pass 2	102	00:00:01.14	00:00:04.59
Symbol table output	50	00:00:00.23	00:00:00.43
Psect synopsis output	2	00:00:00.02	00:00:00.02
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	555	00:00:07.43	00:00:37.72

The working set limit was 1500 pages.

42445 bytes (83 pages) of virtual memory were used to buffer the intermediate code.

There were 40 pages of symbol table space allocated to hold 652 non-local and 38 local symbols.

471 source lines were read in Pass 1, producing 22 object records in Pass 2.

25 pages of virtual memory were used to define 23 macros.

```
+-----+
! Macro library statistics !
+-----+
```

Macro library name

\$255\$DUA28:[SHRLIB]SUM.MLB:1
-\$255\$DUA28:[MACRO.OBJ]MACRO.MLB:1
-\$255\$DUA28:[SYSLIB]STARLET.MLB:2
TOTALS (all libraries)

Macros defined

	Macros defined
\$255\$DUA28:[SHRLIB]SUM.MLB:1	3
-\$255\$DUA28:[MACRO.OBJ]MACRO.MLB:1	12
-\$255\$DUA28:[SYSLIB]STARLET.MLB:2	9
TOTALS (all libraries)	24

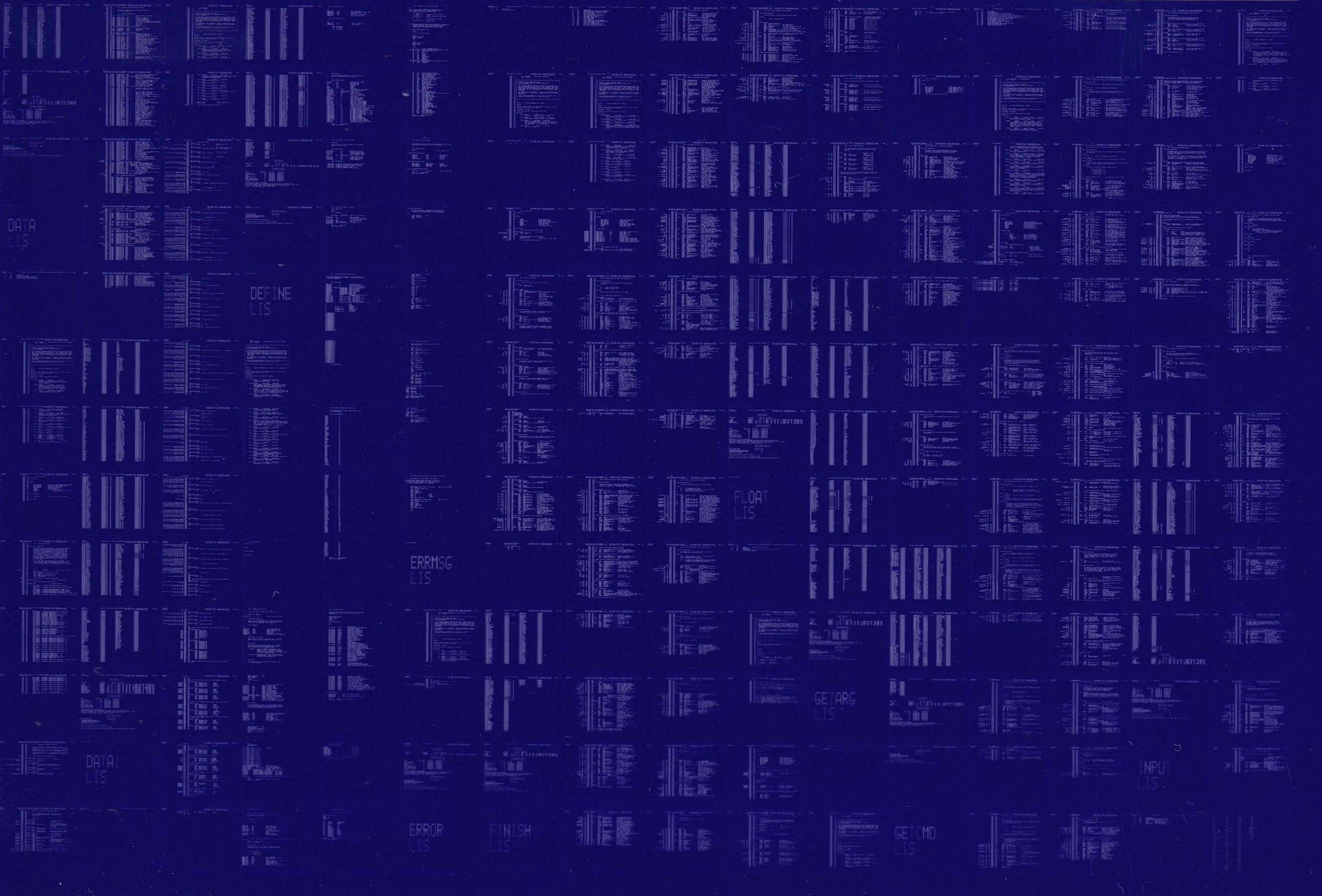
827 GETS were required to define 24 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS\$:\$INPUT/OBJ=OBJ\$:\$INPUT MSRC\$:\$INPUT/UPDATE=(ENH\$:\$INPUT)+LIB\$:\$MACRO/LIB+SHRLIB\$:\$SUM/LIB

0225 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY



0226 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

